Security authentication extension

SuperMap iServer provides the ExtendedUserStorage interface to extend the verification of username and password. Users can make iServer butt joint the current certificate server through extension.

ExtendedUserStorage interface introduction

com.supermap.services.security.ExtendedUserStorage interface has following methods:

- boolean isValid(String username, String password);

It will call this method when need to verify the user name and password in iserver.

- ExtendedUserInfo getUser(String username);

This method is used to get the users' information, which will import the current information into iserver system. These infomation include: groups, roles, description and so on.

When user implements the ExtendedUserStorage interface, it needs to add @ExtendedUserStorageIdentification("extendID") annotation.

Extension and configuration flow

Here we use a simple extension to show the process.

1. Implement a extended user storage class

Implement the ExtendedUserStorageSample class and inherit the ExtendedUserStorage interface. Add the ExtendedUserStorageIdentificatio note as follows:

```
package com.supermap.sample.security;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Map;
import org.apache.commons.lang3.StringUtils;
import com.supermap.services.security.ExtendedUserInfo;
import com.supermap.services.security.ExtendedUserStorage;
import com.supermap.services.security.ExtendedUserStorageIdentification;
@ExtendedUserStorageIdentification("userStorageSample")
public class ExtendedUserStorageSample implements ExtendedUserStorage{
   //Use "exUser,exPassword,exInfo,ADMIN" to represent an existing user
   //Initialization properties can be configured in Jar:///security/extendedUserStorageConfig.xml.
   private String initInfo;
   private Map<String,String> userpasswords = new HashMap<String,String>();
   private Map<String,ExtendedUserInfo> userinfos = new HashMap<String,ExtendedUserInfo>();

   @Override
   public boolean isValid(String username, String password) {
      if(userpasswords.keySet().contains(username)&&userpasswords.get(username).equals(password)){
         return true;
      }else{
         return false;
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

```
        }
    }
    @Override
    public ExtendedUserInfo getUser(String username) {
        if(userinfos.keySet().contains(username)){
            return userinfos.get(username);
        }else{
            return null;
        }
    }
    private void init(){
        String[] strs = StringUtils.split(initInfo, ',');
        String tmpName = strs[0];
        userpasswords.put(tmpName, strs[1]);
        ExtendedUserInfo testUserInfo = new ExtendedUserInfo();
        testUserInfo.description = strs[2];
        testUserInfo.roles = new HashSet<String>();
        testUserInfo.roles.add(strs[3]);
        userinfos.put(tmpName, testUserInfo);
    }
    public void setInitInfo(String initInfo) {
        this.initInfo = initInfo;
        init(); //Initialization
    }
    public String getInitInfo() {
        return initInfo;
    }
}
```

In this example, use initInfo to simulate a user and make a simple initialization. Username: exUser. Password: exPassword. Description: exInfo Role: ADMIN. @ExtendedUserStorageIdentification("userStorageSample") ID: userStorageSample.

2. Configure the extended user storage class

Create the security/extendedUserStorageConfig.xml to configure:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:util="http://www.springframework.org/schema/util"
      xsi:schemaLocation="
      http://www.springframework.org/schema/beans          http://www.springframework.org/schema/beans/spring-beans.xsd
      http://www.springframework.org/schema/util          http://www.springframework.org/schema/util/spring-util-2.5.xsd">
            <bean id="userStorageSample" class="com.supermap.sample.security.ExtendedUserStorageSample">
                <property name="initInfo" value="exUser,exPassword,exInfo,ADMIN"/>
            </bean>
</beans>
```

You can implement the ExtendedUserStorageSample class results, and package the extendedUserStorageConfig.xml to a Jar, placed in %SuperMap iServer_HOME%/webapps/iserver/WEB-INF/lib. Restart the SuperMap iServer services.

Notes: extendedUserStorageConfig.xml locates in Jar:///security/extendedUserStorageConfig.xml. The name

Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

and position of extendedUserStorageConfig.xml can not be modified.

3. View the extension result

Access the service list, namely, http://localhost:8090/iserver/services. Click the right top to login, and input above user extension storage implement. We simulate the user exUser and password exPassword. Click the OK to login. Click exUser to view the details, all role information of exUser.

Here we login through the administrator account. You can also find the exUser in the user tab of iServer.

Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com